Flugg
Ancestor Fun

Ryan Nichol

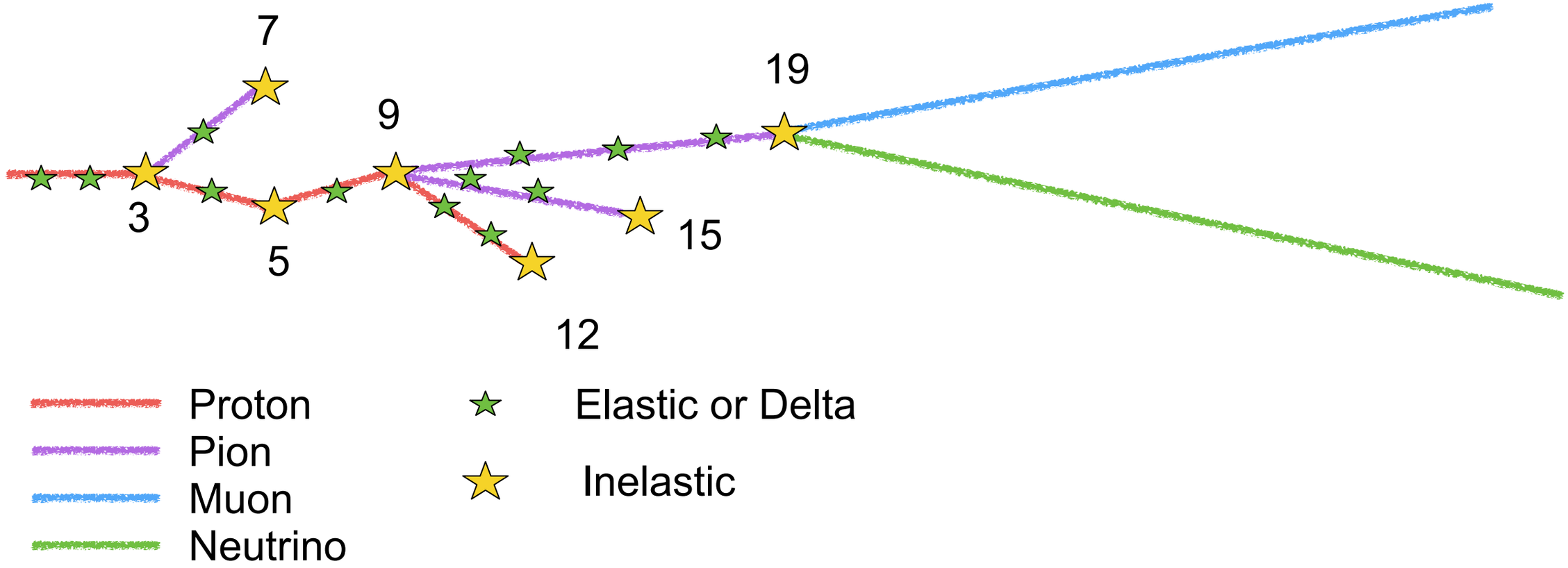
Outline

- In Minos DocDb-11455-v1 we showed that the position of the starting pion position was very different between Flugg and G4NuMI (Minerva V6)
- It turns out that this was down to a bug in the ancestor tracking in Flugg
- A new version has been created to fix this
- In summary:
  - The new method assigns particles to their parents as they are created
  - The old method looped through the list of interactions to try and do the matching only when a neutrino was created

The Old Method

- For each proton on target:
  - Loop over all of the Fluka interactions and store incoming particle type, position and momentum in a temporary array
    - Each particle in theory has a unique fluka particle number
    - The majority of interactions are ones that produce low energy electromagnetic particles
    - Does not store the interaction or decay products
  - Once we find a particle that decays producing a neutrino, need to search back through the temporary arrays to find the ancestors
    - Trivial to trace back along the pion path and then have to ‘guess’ which parent interaction might have produced the pion

Cartoon — Old Method



Each proton on target results in multiple interactions and many daughter particles

We know that at interaction number 19 a pion decayed and produced a neutrino

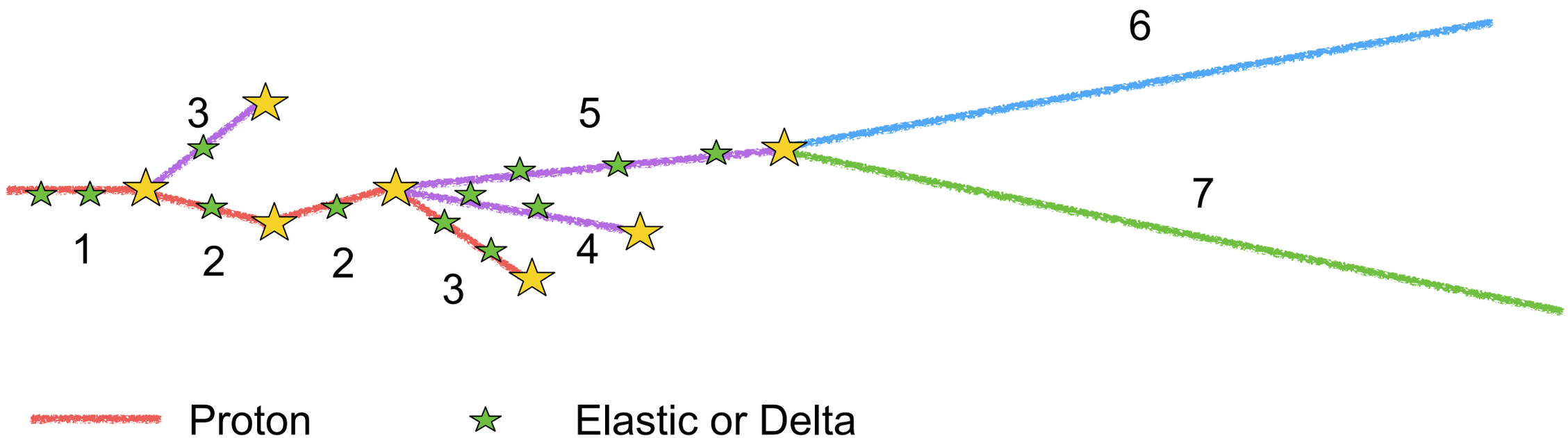
We can unambiguously associate the pion with its elastic or delta interactions, but do not know unambiguously when it was created

So it is possible to mistake the pion that decayed at point 19 with the one that was created at interaction point 3

The New Method

- For each proton on target:
  - Give each particle a unique number, incoming proton=0
  - Loop over the interactions
    - Check if the interacting particle is the same as the previous particle
    - At each inelastic interaction or decay loop over the daughter list and
      - Record unique particle number, unique parent particle number, start position and start momentum
  - If the interacting particle does not match the last particle loop through the list of produced particles and match the particle type and z-momentum
  - If a neutrino is produced already have the fully connected ancestor chain

Cartoon — New Method



The only complication is that when fluka starts tracking a new particle we have to match that particle to its creation point

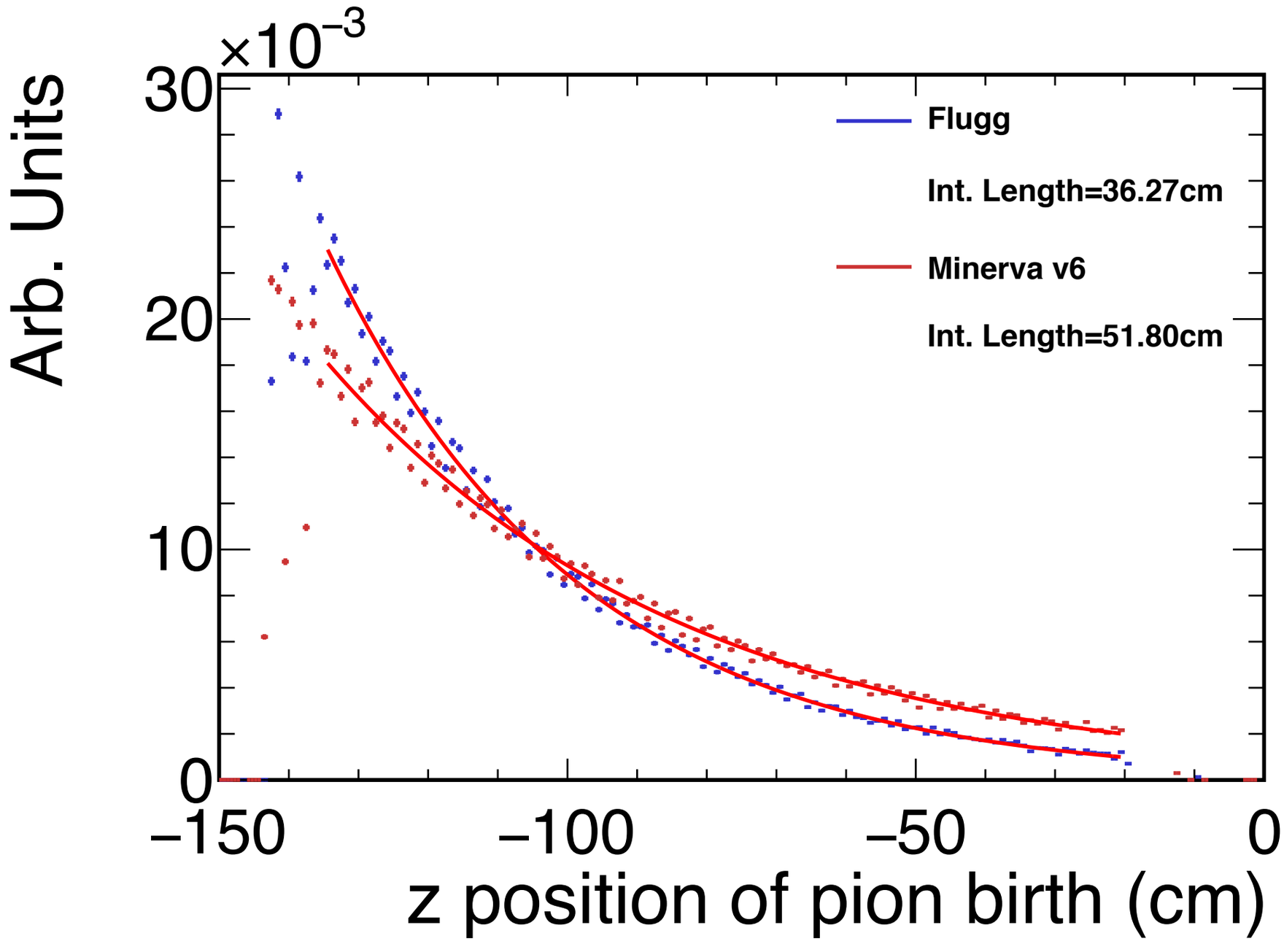
We do this by picking the particle (i.e. pion) that had the closest z-momentum

From daughter list at each interaction:

Proton 1 —> Proton 2 + Pion 1
Proton 2 —> Proton 3 + Pion 4 + Pion 5
Pion 5 —> Muon 6 + Neutrino 7

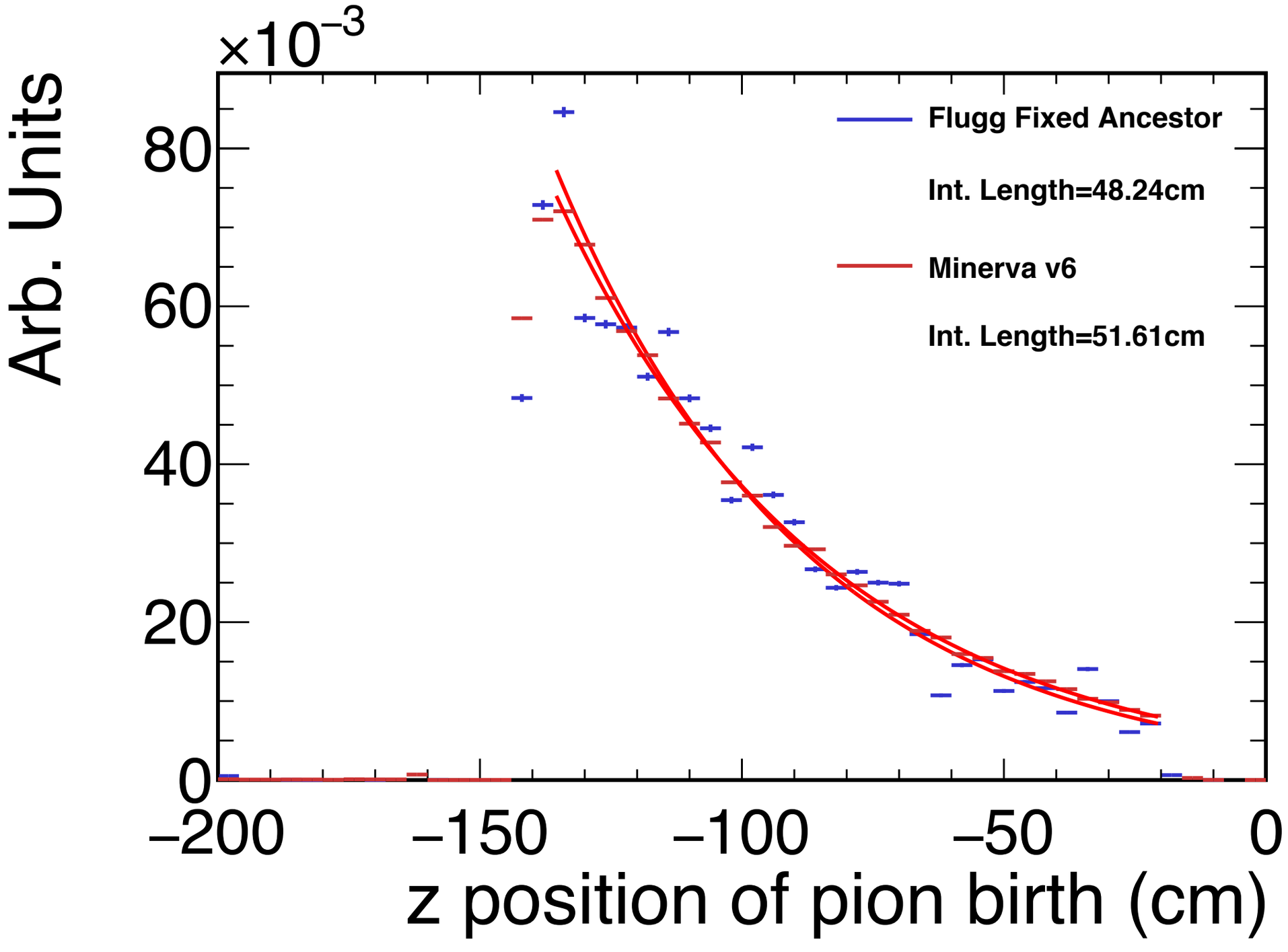
Broken Matching — Pion Start Position

- The initial pion starting position looked wrong in Flugg… it was wrong but only due to the ancestor list bug



Fixed Matching — Pion Start Position

- The stats are very low but the new ancestor matching ‘fixes’ the starting pion position plot



Summary

- The ancestor matching was broken in Flugg.
- We have fixed the ancestor matching basically by moving to a system where we associate particles on the fly rather than trying to do matching once we have a neutrino
- As far as I can tell this didn't break anything or change the flux at the detectors
- The change has been committed to the repository

```
cvs commit -m "Adding a fix to the ancestor matching. See numix-doc-125 for more details"
cvs commit: Examining .
/cvs/projects/numi-beam-sim/numi-beam-sim/g4numi_flugg/for/(NUMI),v  <--  (NUMI)
new revision: 1.3.6.2; previous revision: 1.3.6.1
/cvs/projects/numi-beam-sim/numi-beam-sim/g4numi_flugg/for/Attic/mgdraw.f.minos,v  <--  mgdraw.f.minos
new revision: 1.1.2.5.2.6; previous revision: 1.1.2.5.2.5
/cvs/projects/numi-beam-sim/numi-beam-sim/g4numi_flugg/for/Attic/mgdraw.f.nova,v  <--  mgdraw.f.nova
new revision: 1.1.2.7.2.6; previous revision: 1.1.2.7.2.5
cvs commit: Using deprecated info format strings.  Convert your scripts to use
the new argument format and remove '1's from your info file format strings.
```